DOCUMENT RESUME

ED 108 645 IR 002 168

AUTHOR Kline, Lanaii

TITLE Song File Match.
INSTITUTION Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.
REPORT NO SWRL-TN-5-72-63
PUB DATE 20 Dec 72
NOTE 22p.

EDRS PRICE MF-$0.76 HC-$1.58 PLUS POSTAGE
DESCRIPTORS *Computer Programs; *Data Bases; Data Processing; *Design; *Electronic Data Processing; Flow Charts; Information Retrieval; Information Storage; Information Systems; Music; Systems Development; *Vocal Music
IDENTIFIERS Computer Software Documentation; FORTRAN; *SONG FILE DATA MATCH; Songs

ABSTRACT

Computer routines designed to determine if a song already exists in a song file data base are described. Written in FORTRAN IV, SONG FILE DATA MATCH compares the note sequence of songs in the data base and the one to be added to it. Information is then presented to the user which analyzes the sequence of notes and compares them to other songs. This document is intended to serve as the software documentation for the programs. Included are a program description, data format specifications, program constraints and limitations, and operating instructions. Program flowcharts, program listings, and sample data forms and reports are also provided. (DGC)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED108645

JR

SWRL

SOUTHWEST REGIONAL LABORATORY TECHNICAL NOTE

DATE: December 20, 1972

NO: TN 5-72-63

TITLE: SONG FILE MATCH

AUTHOR: Lanaii Kline

ABSTRACT

Song File Match is a computer program which compares the note sequence of a new entry to those previously stored in the song file database. Songs containing matching sequences are listed.

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

JR 082 168

This document is intended for internal staff distribution and use. Permission to reprint or quote from this working document, wholly or in part, should be obtained from SWRL, 4665 Lampson Avenue, Los Alamitos, California 90720

1.0 - PROGRAM IDENTIFICATION

SONG FILE NOTE MATCH

2.0 - OBJECTIVE

To check if a song already exists in the song file database by comparing the note sequence of the songs in the database and the one to be added to the database later.

3.0 - PROGRAM DESCRIPTION

3.1 - Program Logic

The Song File database is read and sorted by note sequence (twelve notes). A song title and note sequence is read. The program searches the Song File for a match on the note sequence. All matches are printed. This process continues until all songs to be checked have been processed.

3.2 - Variables and Internal Tables

Integer Arrays:

IHEAD (45) - is a scratch file used for the sort subroutine.

LIST (45) - contains the indices of the array which was sorted in the order of the sort.

NOTE (25) - contains the input record which is to be checked for duplication in the Song File.

POINT (45) - contains the indices of LIST.

SONG (45, 59) - contains the Song File in core.

Integers:

COMPAR - is an input/output argument for the functions SORT and COMPAR.

M	- is the denominator for finding the range of records to search in the database.
S	- is the smallest mean be.
UB	- is the upper bound of the portion of the song file to be searched.
LOW	- is the lower bound of the portion of the song file to be searched.
P	- is the index of the note in the records of the song file and the input array which are to be compared for a match.
NUMB	- in the maximum number of elements in the song file to be searched. 1 NUMB N.
N	- is the number of records in the song file.
MID	- is the smallest range of records used in searching for a match.
OLD	- is the old lower bound.
OLDER	- is the old upper bound.
I	- is the index for LIST.
M	- is the index for POINT.
RANGE	- is the range of records to be searched in the Song File. RANGE = UP - LOW.
BOTTOM	- is the lower portion of the array which is not to be checked. When P = 14 then BOTTOM = 0; when P = 15, then BOTTOM = LOW.

5.0 - DATA DESCRIPTION

5.1 - Input Formats

See Attachment 1

5.2 - Output Formats

See Attachment 1

6.0 - PROGRAM CONSTRAINTS AND LIMITATIONS

6.1 - Programming Language

FORTRAN IV

6.2 - Vendor

University of California, Los Angeles Campus Computing Network.

6.3 - Storage Requirements

Compilation: 98K

Execution: 56K

I/O Requests: 361

6.4 - Hardware Configuration

IBM 360 Model 91, Disk, Card Reader, Line Printer

6.5 - Program Parameters

The program is currently limited to forty-five records in song file. To increase this number change the following cards:

DIMENSION NOTE (25), LIST (45)*, 1 HEAD (45)*, POINT (45)*

COMMON /X1/ SONG (45, 59)*

1F (N.GT.45) Go to 23

in Integer function COMPAR

COMMON /X1/ SONG (45, 59)*

6.6 - Error Messages

EXCEEDED THE SONG ARRAY

Processing terminates

7.0 - OPERATING INSTRUCTIONS

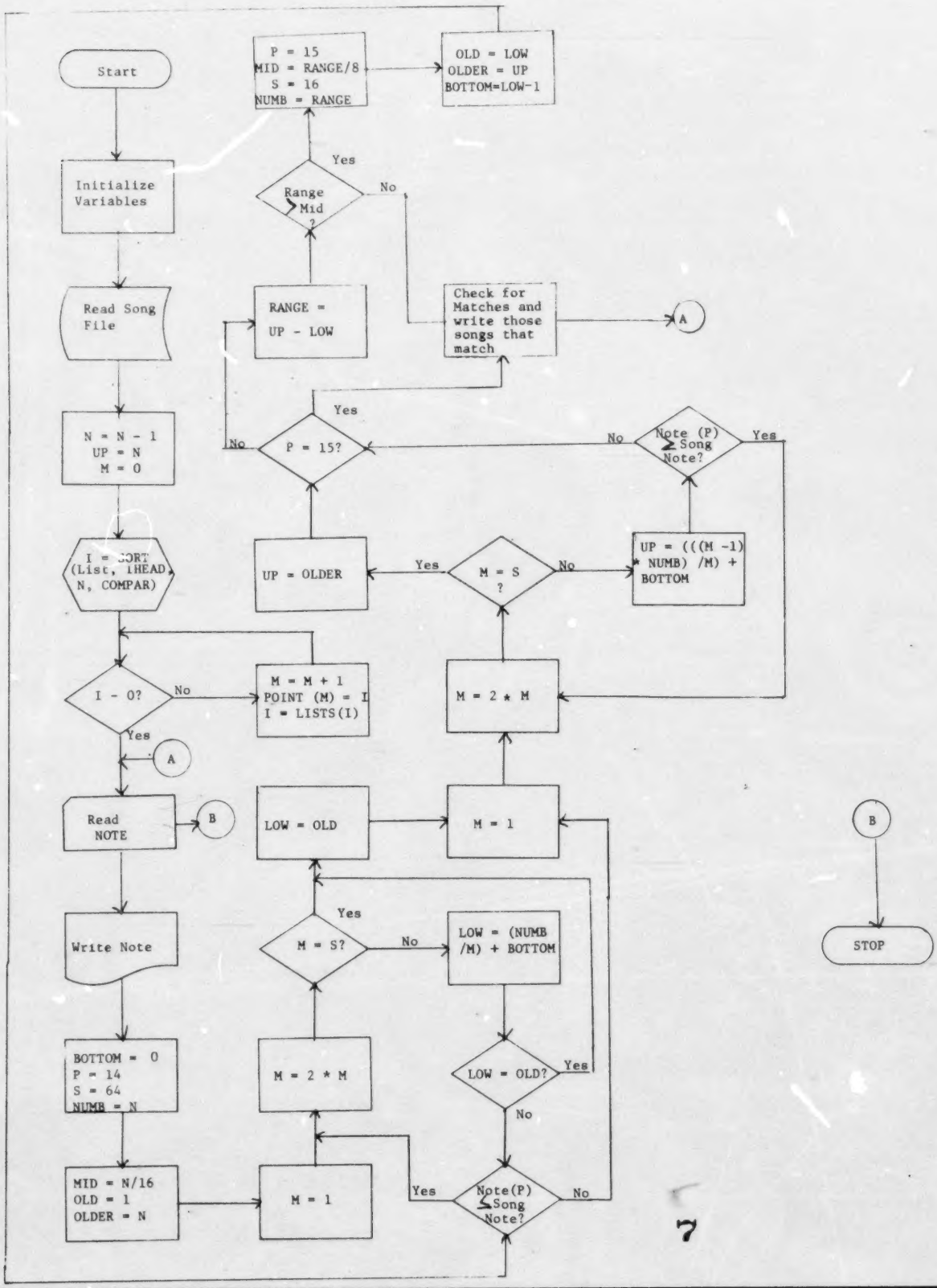
Run in batch mode

```
// job card
// password
// EXEC FORTGCLG
// FORT. SYSIN DD *
   program deck
/ *
// GO. FT04F001 DD DSN =
// GO. SYSIN DD *
   data deck
/ *
//
```

SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanaii Kline
PROGRAM ID: SONG FILE MATCH
DATE:
PAGE 4 OF 1



9.0 - PROGRAM LISTING

```
      IMPLICIT INTEGER (A-Z)
      DIMENSION NOTE(25),LIST(45),IHEAD(45),POINT(45)
      COMMON /X1/SONG(45,59)
      EXTERNAL COMPAR
C
C       M      - DENOMINATOR FOR FINDING SEARCH RANGE TO PARTITION THE
C                DATABASE
C       S      - EQUALS THE SMALLEST DENOMINATOR USED TO PARTITION THE
C                SONG FILE ARRAY
C       UP     - UPPER BOUND OF PORTION OF SONG FILE TO BE SEARCHED
C       LOW    - LOWER BOUND OF PORTION OF SONG FILE TO BE SEARCHED
C       P      - INDEX OF NOTE IN NOTE SEQUENCE ARRAY
C       NUMB   - NO. OF ELLEMENTS IN THE SONG FILE ARRAY
C       N      - NO. OF ELLEMENTS IN THE SONG FILE ARRAY
C       MID    - THE SMALLEST RANGE OF ELEMENTS TO BE PARTITIONED FROM
C                THE SONG FILE ARRAY
C       CLD    - THE CLD LOWER BOUND
C       CLDER  - THE OLD UPPER BOUND
C
      WRITE(6,10)
10    FORMAT('1')
      N=1
      LOW=1
C
C     READ THE SONG FILE
C
3     READ(4,1,END=2) (SONG(N,I),I=1,59)
1     FORMAT(5X,12A4,A3,12I2/5X,18A4,A3/I5,4X,13A4,A3)
      N=N+1
      IF(N.GT.45) GO TO 23
      GO TO 3
C
C     END READING SONG FILE - SORT NOTE SEQUENCES
C
2     N=N-1
      UP=N
      M=0
      I=SORT(LIST,IHEAD,N,COMPAR)
5     IF(I.EQ.0) GO TO 4
      M=M+1
      POINT(M)=I
      I=LIST(I)
      GO TO 5
C
C     READ INPUT CARD
C
4     READ(5,6,END=7) (NOTE(J),J=1,25)
6     FORMAT(5X,12A4,A3,12I2)
      WRITE(6,8) (NOTE(J),J=1,25)
8     FORMAT('0'////'0',12A4,A3,10X,'NOTE SEQUENCE -',1X,12I2)
      BOTTOM=0
      P=14
      S=64
      NUMB=N
      MID=N/16
```

```
      OLD=1
      OLDER=M
C
C     TEST FOR LOWER BOUND : LOW
C
20    M=1
9     M=2*M
      IF(M.EQ.S) GO TO 11
      LOW=(NUMB/M)+BOTTOM
      IF(LOW.LT.OLD) GO TO 11
      IF(NOTE(P).LE.SONG(POINT(LOW),P)) GO TO 9
C
C     NOTE GREATER THAN LOW
C     FOUND LOWER BOUND
C
C     TEST FOR UPPER BOUND : UP
C
18    M=1
12    M=M*2
      IF(M.EQ.S) GO TO 19
      UP=(((M-1)*NUMB)/M)+BOTTOM
      IF(NOTE(P).GE.SONG(POINT(UP),P)) GO TO 12
C
C     NOTE LESS THAN UP
C     FOUND UPPER BOUND
C
22    IF(P.EQ.15) GO TO 21
      RANGE=UP-LOW
      IF(RANGE.GT.MID) GO TO 13
C
C     TEST IF NOTE IS IN RANGE
C
21    DO 14 I=LOW,UP
      DO 15 J=14,25
      IF(NOTE(J).NE.SONG(POINT(I),J)) GO TO 14
C
C     NOTES MATCH SO FAR
C
15    CONTINUE
C
C     NOTES MATCH NOTES OF SONG THE DATABASE
C
      WRITE(6,16) SONG(POINT(I),45),(SONG(POINT(I),J),J=1,44)
16    FORMAT('0THE NOTE SEQUENCES MATCH THAT OF SONG NUMBER',1X,I5/' ',
     +10X,12A4,A3,10X,'SEQUENCE -',1X,12I2/' ',10X,18A4,A3)
14    CONTINUE
C
C     NO MATCH
C
      WRITE(6,17)
17    FORMAT('0NO MORE MATCHES FOR THIS NOTE SEQUENCE')
      GO TO 4
C
C     LOWER BOUND IS 1 OF BOTTOM OF RANGE
C
```

```
11    LOW=OLD
      GO TO 18
C
C     RANGE IS TOO LARGE
C
13    P=15
      MID=RANGE/8
      S=16
      NUMB=RANGE
      OLD=LOW
      OLDER=UP
      BOTTOM=LOW-1
      GO TO 20
C
C     LOWER BOUND IS N OR TOP OF RANGE
C
19    UP=OLDER
      GO TO 22
C
C     READ MORE DATA THAN ARRAY WAS DIMENSIONED
C
23    WRITE(6,24)
24    FORMAT('1 EXCEEDED THE ARRAY SONG')
C
C     END OF FILE
C
7     WRITE(6,10)
      STOP
      END
```

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

TITLE: Database - Song File DATE: ____

FILE ID: Input

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED ■
LENGTH 80
VARIABLE ☐
MAX ____
MIN ____

BLOCKING FACTOR:
BLOCK CONTAINS
91 RECORDS

REMARKS:

Output file uses same format as the input file.

CARD LAYOUT ☐
TAPE RECORD LAYOUT ■

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D.
Title/Note Sequence

FORMAT	A	A																	
POSITIONS	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

TITLE

NOTE SEQUENCE (12 Notes)

☐ CONTINUED

RECORD I.D.
Composer/Publisher

FORMAT	△	A																	
POSITIONS	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

COMPOSER/PUBLISHER

☐ CONTINUED

11 RECORD I.D.
Song Characteristics

FORMAT	N	△	N	N	N	A	N	A	N	A	N	A	N	A	N	N	N	△	
POSITIONS	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

Song Number

melody

mode

range

Melodic Features

Rhythmic Features

Structural Features

Harmonic Features

seasons

holidays

American

Non-American

☐ CONTINUED

RECORD I.D.

FORMAT																			
POSITIONS	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

Page ___ of

TITLE: Match Record DATE:

FILE ID: Input

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED ■
LENGTH 80
VARIABLE ☐
MAX
MIN

BLOCKING FACTOR:
BLOCK CONTAINS
☐ RECORDS

REMARKS:

CARD LAYOUT ■
TAPE
RECORD LAYOUT ☐

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D.

FORMAT	A	TITLE																	NOTE Sequence (12 notes)	
POSITIONS	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

☐ CONTINUED

RECORD I.D.

FORMAT																				
POSITIONS	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

☐ CONTINUED

12 RECORD I.D.

FORMAT																				
POSITIONS	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

☐ CONTINUED

RECORD I.D.

FORMAT																				
POSITIONS	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

```
      INTEGER FUNCTION CCMPAR(I,J)
      IMPLICIT INTEGER (A-Z)
      COMMON /X1/SONG(45,59)
      K=14
40    IF(SONG(I,K)-SONG(J,K)) 10,20,30
10    CCMPAR=-1
      RETURN
30    CCMPAR=1
      RETURN
20    CCMPAR=0
      K=K+1
      IF(K.LE.25) GO TO 40
      RETURN
      END
```

```
      INTEGER FUNCTION SORT(LISTS,IHEAD,N,COMPAR)
C
C     GENERAL SORT ROUTINE
C     LISTS IS AT LEAST N WORDS LONG, WHERE N IS THE NUMBER OF RECORDS
C     TO BE SORTED.
C     IHEAD IS ALSO N WORDS LONG AND IS USED AS A SCRATCH AREA.
C     COMPAR IS A 2-ARGUMENT INTEGER FUNCTION HAVING A NEGATIVE VALUE
C     IF THE RECORD REFERENCED BY THE FIRST ARGUMENT IS TO BE SORTED
C     BEFORE THE RECORD REFERENCED BY THE SECOND ARGUMENT, A ZERO VALUE
C     IF THE ORDER IS IMMATERIAL, AND A POSITIVE VALUE OTHERWISE.
C     SINCE SORT DEALS WITH RECORD INDEXES ONLY, THE SORT KEYS MAY BE
C     ARBITRARILY COMPLEX, AND THE RECORDS TO BE SORTED NEED NOT BE
C     IN MEMORY.
C
C     USAGE
C
C     INTEGER SORT
C     EXTERNAL COMPAR (OPTIONAL, COMPAR MAY BE INTERNAL)
C     I=SORT(LIST,IHEAD,N,COMPAR)
C   A IF (I.EQ.0) GO TO B
C     DO ANY PROCESSING FOR RECORD I HERE.
C     I=LIST(I)
C     GO TO A
C   B CONTINUE
C
      IMPLICIT INTEGER (A-Z)
      INTEGER COMPAR,DIRECT
      INTEGER LISTS(N),IHEAD(N)
      EQUIVALENCE (MP,DIRECT),(JHEAD,NCHAIN)
      EXTERNAL COMPAR
C
C     START BY SCANNING ALL RECORDS AND DETECTING ASCENDING OR
C     DESCENDING CHAINS.  PUT THE HEADS OF THESE CHAINS IN HEAD.
C
      M=1
C
C     INDICATE BEGINNING OF CHAIN
C
      DIRECT=0
      I=1
C
C     INDICATE RECORD 1 HEAD OF CHAIN
C
      JHEAD=1
      LISTS(1)=0
1     IF(I.GE.N) GO TO 10
      I1=I+1
C
C     CHECK SEQUENCE OF TWO ADJACENT RECORDS.
C
      IF(COMPAR(I,I1)) 2,22,8
22    IF(DIRECT) 9,3,4
C
C     RECORDS I,I+1 ARE IN ASCENDING SEQUENCE.
C
```

```
2       IF(DIRECT) 7,3,4
C
C       START A FORWARD CHAIN.
C
3       DIRECT=1
        JHEAD=I
C
C       CONTINUE A FORWARD CHAIN.
C
4       LISTS(I)=I1
5       I=I1
        GO TO 1
C
C       HAD A CHAIN, GOT A SEQUENCE CHANGE.
C       INDICATE END OF CHAIN
C
6       LISTS(I)=0
C
C       SAVE HEAD OF CHAIN
C
7       IHEAD(M)=JHEAD
        M=M+1
        DIRECT=0
        GO TO 5
C
C       RECORDS I,I+1 ARE IN DESCENDING SEQUENCE.
C
8       IF(DIRECT) 9,99,6
C
C       START A BACKWARD CHAIN.
C
99      DIRECT=-1
C
C       INDICATE END OF CHAIN
C
        LISTS(I)=0
C
C       CONTINUE A BACKWARD CHAIN
C
9       LISTS(I1)=I
C
C       UPDATE HEAD OF CHAIN
C
        JHEAD=I1
        GO TO 5
C
C       END OF CHAIN DETECTION.  CLEAN UP AND START MERGE PASS.
C
10      IHEAD(M)=JHEAD
        IF (DIRECT.GE.0) LISTS(I)=0
        IF (DIRECT.EQ.0) IHEAD(M)=N
11      I=-1
        MP=0
12      I=I+2
        IF (I.LT.M) GO TO 15
```

```
      IF (I.EQ.M) GO TO 14
13    M=MP
      IF (M.NE.1) GO TO 11
      SORT=IHEAD(1)
      IHEAD(1)=0
      RETURN
14    MP=MP+1
      IHEAD(MP)=IHEAD(I)
      IF (I.EQ.1) GO TO 13
      IHEAD(I)=0
      GO TO 13
15    J1=IHEAD(I)
      J2=IHEAD(I+1)
      IHEAD(I)=0
      IHEAD(I+1)=0
      I1=J1
      I2=J2
      MP=MP+1
      NCHAIN=0
16    IF (COMPAR(J1,J2).LE.0) GO TO 17
C
C     J2 SHOULD COME BEFORE J1.
C
      IF (NCHAIN.EQ.0) IHEAD(MP)=J2
      IF(NCHAIN.EQ.1) LISTS(I1)=J2
      NCHAIN=2
      I2=J2
      J2=LISTS(I2)
      IF(J2.NE.0) GO TO 16
      LISTS(I2)=J1
      GO TO 12
C
C     J1 SHOULD COME BEFORE J2.
C
17    IF (NCHAIN.EQ.0) IHEAD(MP)=J1
      IF (NCHAIN.EQ.2) LISTS(I2)=J1
      NCHAIN=1
      I1=J1
      J1=LISTS(I1)
      IF(J1.NE.0) GO TO 16
      LISTS(I1)=J2
      GO TO 12
      END
```

ATTACHMENT 3
SAMPLE INPUT

THE BUNNY HOP	12121214101010101015121 2
THE MEXICAN HAT DANCE	18161816151418201914161 7
THE WASSAIL SONG	18181515151212141510101 0
MY HAT IT HAS THREE CORNERS	18181616171715151414141 8
OH WHERE IS THAT DOGGY IN THE WINDOW	14171812131518161718121 2
THE CHIPMUNK SONG	18181617151213141718122 0
LONDON BRIDGES FALLING DOWN	18181816171818191915141 3
HERE WE GO ROUND THE MULBERRY BUSH	13121415161718192021131 5

```
                                                                    181618161514182019141617
MARY HAD A LITTLE LAMB
FOLK SONG                                    1                        1 301
1     11  1   1                                                  1312141516171819202113 15
WHITE CORAL BELLS
GIRL SCOUT SONG                                          2 420
2    225      1   1                                     1818181617181819191514 13
THREE BLIND MICE
CHILDREN'S SONG                              1                      1 205
3    31      1                                          181816171512131417181220
ROW ROW YOUR BOAT
ROUND                                                               11001
4    522 1              1          1                   181816161717151514141418
GREEN GROW THE LILACS
FOLK SONG                                    1                      1 120
5    111   1                                          181815121416 3 4  8 9 112
RAINDROPS KEEP FALLING ON MY HEAD
BACHARACH                                    1                      2 340
6    11  1   1                                       1818191917172020212122 22
YELLOW SUBMARINE
BEATLES                                                             1 303
7    225      1   1   1          1                   121212121212 8 1 2 6 9 3
MICHELLE
BEATLES                                                             11020
8    31      1          1                            181819192021222524271416
BLOWIN IN THE WIND
PETER, PAUL AND MARY                                                11010
9    111                                             1812171818181818151517 17
IT HAD TO BE YOU
UNKNOWN                                                             3 330
10    12      1    1                                 151716121413171815121416
TEA FOR TWO
ANONYMOUS                                                           1 202
11    11   1          1          11      1            121212121417171414171717
SILENT NIGHT
FRANZ GRUBER                                                        1 220
12    111 1                                          1818181818151515151714 14
THE YELLOW ROSE OF TEXAS
ANONYMOUS                                                           2 450
13    1 1  1                                         1818181717141516121718 19
STAR SPANGLED BANNER
FRANCES SCOTT KEY                                                   1 110
14    1     1                                       1818151717171819191820 21
JINGLE BELLS
CHRISTMAS                                                           1 130
15    056   111                                     181618161514182019141617
TIPTOE THROUGH THE TULIPS
TINY TIM                                                    1       1 1 01
16    111 111                                        131214151617181920211315
MARY HAD A LITTLE LAMB
CHILDREN'S NURSERY RHYME                     1          1       1   1 1 220
17    211                                            1818181617181819191514 15
KUMBYYA MY LORD
AFRICAN FOLK SONG                            1          1           1 1   31005
18    312      1                                     201816171512131417181220
SHALOM CHAVERIM
```

```
  HEBREW CHANNUKAH SONG
19    053            1          1       1       1      1 3 210
  SUNRISE SUNSET                                      181812171517151417181912
  FIDDLER ON THE ROOF
20               1                1                     1 1 10
  GETTING TO KNOW YOU                                 181815171812101417191919
  ROLGERS AND HAMMERSTEIN
21    111                                                1 710
```

THE BUNNY HOP — NOTE SEQUENCE - 12121214101010101012121?

NO MORE MATCHES FOR THIS NOTE SEQUENCE

THE MEXICAN HAT DANCE — NOTE SEQUENCE - 181618161514182019141617

THE NOTE SEQUENCES MATCH THAT OF SONG NUMBER 1
MARY HAD A LITTLE LAMB — SEQUENCE - 181618161514182019141617
POLK SONG

THE NOTE SEQUENCES MATCH THAT OF SONG NUMBER 16
TIPTOE THROUGH THE TULIPS — SEQUENCE - 181618161514182019181617
TINY TIM

NO MORE MATCHES FOR THIS NOTE SEQUENCE

THE HASSAN SONG — NOTE SEQUENCE - 181815151512121415101010

NO MORE MATCHES FOR THIS NOTE SEQUENCE

20

MY HAT IT HAS THREE CORNERS — NOTE SEQUENCE - 181816161717151514141418

THE NOTE SEQUENCES MATCH THAT OF SONG NUMBER 5
GREEN GROW THE LILACS — SEQUENCE - 181816161717151514141418
POLK SONG

NO MORE MATCHES FOR THIS NOTE SEQUENCE

OH WHERE IS THAT FOGGY IN THE WINDOW — NOTE SEQUENCE - 141718121315181617141217

NO MORE MATCHES FOR THIS NOTE SEQUENCE

THE CHIMNEY SONG

NOTE SEQUENCE - 1818161715121314171812?0

THE NOTE SEQUENCES MATCH THAT OF SONG NUMBER 4
ROW ROW YOUR BOAT
ROUND

SEQUENCE - 1818161715121314171812?0

NO MORE MATCHES FOR THIS NOTE SEQUENCE

LONDON BRIDGES FALLING DOWN

NOTE SEQUENCE - 1818181617181819191514?3

THE NOTE SEQUENCES MATCH THAT OF SONG NUMBER 3
THREE BLIND MICE
CHILDREN'S SONG

SEQUENCE - 1818181617181819191514?3

NO MORE MATCHES FOR THIS NOTE SEQUENCE

HERE WE GO ROUND THE MULBERRY BUSH

NOTE SEQUENCE - 1312141516171819202?1315

THE NOTE SEQUENCES MATCH THAT OF SONG NUMBER 2
WHITE CORAL BELLS
CARL SCOUT SONG

SEQUENCE - 1312141516171819202?1315

THE NOTE SEQUENCES MATCH THAT OF SONG NUMBER 17
MARY HAD A LITTLE LAMB
CHILDREN'S NURSERY RHYME

SEQUENCE - 1312141516171819202?1315

NO MORE MATCHES FOR THIS NOTE SEQUENCE

DISTRIBUTION LIST

Linda Miller

Lanaii Kline (3)

Frank Teplitzky

Dick Piper

Directorate

Bill Coulton

Library & Archives